

Exhibit

A-10



6/A
2-13-D3
NP
PATENT
Attorney Docket No. 1265

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:
TOEPKE et al.

Group Art Unit: 2674

Serial No.: 10/072,111

Examiner: NGUYEN, K.

Filed: February 8, 2002

For: SOFT INPUT PANEL SYSTEM
AND METHOD

AMENDMENT AND RESPONSE TO OFFICE ACTION

FEB 11 2003

Commissioner for Patents
Washington, D.C. 20231

Technology Center 2600

Dear Sir:

This communication is a response to the Office action mailed July 30, 2002. Please enter the following amendments and consider the appended remarks.

IN THE CLAIMS:

Please cancel claim 22 and 38.

Please amend claims 18, 21, 23-25, 29 and 36 as follows (a marked up copy of the amended claims showing the additions and deletions to the previous version is attached hereto as Appendix A):

a-1
18. (Amended) The system of claim 12 wherein the executable input method comprises a component object model object;

A

In re Application of TOEPKE et al.
Serial No.: 10/072, 111

21. (Amended) A system for receiving user data input into a computer system having a plurality of application programs, comprising:

a plurality of executable input methods, each executable input method being distinct from the application programs and other executable input methods and configured to accept the user data input from an input device associated with the computer system, each executable input method further including a defined interface set including at least one interface therein to make the executable input method pluggable into other executable code that is capable of interfacing with the defined interface set;

A2
an input panel window on a touch-sensitive display screen that is distinct from a window of the application, and wherein the executable input method includes executable code that draws an input panel in the input panel window; and

a management component that is capable of interfacing with the defined interface set, the management component being distinct from the application programs and configured to:

1) identify one of the executable input methods as a selected executable input method,

2) activate the selected executable input method,

3) communicate with the selected executable input method via the defined interface set,

4) identify information about user data received by the selected executable input method, and

5) pass the information about the received user data to an active application program of the plurality of application programs.

In re Application of TOEPKE et al.
Serial No.: 10/072, 111

22
23
23. (Amended) The system of claim 21 wherein the input panel includes an image representing a keyboard having a plurality of keys thereon.

24
25
24. (Amended) The system of claim 21 wherein the management component selectively displays and hides the display of the input panel window.

26
26. (Amended) The system of claim 21 wherein interaction with the input panel does not cause the input panel window to receive focus.

27
28
28. (Amended) The system of claim 21 wherein the executable input method comprises a component object model object.

29
30
30. (Amended) A system for receiving user input data into a computer system having a graphical windowing environment, comprising:

31
32
32. an input-output mechanism including a display screen that outputs images and a touch sensitive input device that detects user interaction therewith;
33
33. a management component operatively connected to the graphical windowing environment and configured to create an input panel window for display thereof by the graphical windowing environment on the screen, the management component further configured to receive user data and communicate the user data to the graphical windowing environment;

In re Application of TOEPKE et al.
Serial No.: 10/072, 111

a plurality of executable input methods, each executable input method including a defined interface set comprising at least one interface for calling functions of the management component, and further including an input panel corresponding thereto; and

*A5
CMD*
a selected executable input method of the plurality that:

- 1) draws its corresponding input panel by calling a function of the management component via the defined interface set,
- 2) receives user data input via the input panel, and
- 3) calls a function of the management component via the defined interface set to pass the user data thereto; and

an executable program distinct from the selected executable input method, the graphical windowing environment communicating the user data to the executable program.

REMARKS

The Office action dated July 30, 2002 has been carefully considered. In the Office action, claims 18 and 20 were rejected for indefiniteness, and claims 21-38 were rejected under 35 U.S.C. § 102(b) as being anticipated by Stucka et al, U.S. Patent No. 5,596,702 (hereafter "Stucka"). Claims 1-20 and 39-49 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Stucka in view of Kono, U.S. Patent No. 5,914,707 (hereafter "Kono"), (although a "Mori et al. reference was also mentioned in the rejection). By the present amendment, claims 22 and 38 have been canceled, claims 18, 21, 23-25, 29 and 36 have been amended, and the rejections traversed as explained in the following remarks. Reconsideration is respectfully requested.

In re Application of TOEPKE et al.
Serial No.: 10/072, 111

Considering first the §112 rejections, applicants have spelled out the term “COM” (as “component object model”) in claims 18 and 29. The component object model (COM) is well understood in the art, is described in the specification in a manner consistent with that known in the art, and thus applicants submit that claims 18 and 29 are not indefinite. Reconsideration and withdrawal of the §112 rejections is respectfully requested.

Turning to the rejection on the art, the present invention is generally directed to input methods that are separate from application programs, and provide their own input panels to receive user input in windows that are distinct from the application program’s window or windows. The input methods are interchangeable, user-selectable software components that receive (and possibly transform) data that is then provided to the various application programs. To this end, a user selects an input method, which draws its control panel, and the user then enters data via that input method’s control panel. The received data is often transformed into another type of data that is passed to an active application program. For example, as generally described in the present application, one input method may convert taps on a screen to keyboard characters, while another input method may receive handwriting (and/or speech), and output corresponding text characters based on the handwriting (and/or speech). The active application that receives the data does not need to know anything about how the data was input, what the input method did to transform the data, and so on, and instead only needs to process the data that the input method provided (such as text data transformed from screen taps). Note that the above description is for informational and example purposes only, and should not be used to limit the claims, which are discussed below.

In re Application of TOEPKE et al.
Serial No.: 10/072, 111

In contrast to the present invention, Stucka is directed towards providing *applications* with the ability to dynamically construct their own user interfaces using display components selected from a shared server, via application-issued commands. Stucka, column 9, lines 3-13; see also Stucka, column 24, line 62 to column 28, line 4, for a specific example of how an application constructs its displayed user interfaces. Significantly, in Stucka, “each application program controls the display and appearance of their user interfaces by issuing commands” to a server. Stucka, column 16, lines 51-53.

Stucka’s teachings are thus directly *opposite* the fundamental concept of an input method as defined and claimed in the claims of the present invention. For one, the input methods of the present invention are not built and output by the application program, but are independent software entities, that among other things, essentially draw themselves into an input method window. Note that the claims essentially point out that the input methods are distinct from the application programs that ultimately receive the data provided by the input methods. For another, the input methods of the present invention are interchangeable, user selectable and/or controllable, unlike the user interfaces of the application program in Stucka, in which the application program chooses and builds the user interfaces that the application program needs.

In direct contrast to Stucka, with data received at an input method, the initial user interface is essentially external to the application’s user interface, e.g., the input method gets the input data via its own control panel, transforms the input data in some way, and then provides it (e.g., via a message queue) to the application’s user interface, as if the data (in its transformed state) was received directly by the application’s user interface. Indeed, one of the primary benefits of the present invention is that virtually any input method can work with virtually any

In re Application of TOEPKE et al.
Serial No.: 10/072, 111

application program, even those applications that are completely unaware of input methods and can only handle basic input, such as text or possibly mouse commands.

By law, in order to support an anticipation rejection, the Office action is required to show that each and every element of the claimed invention is disclosed in a single reference, and that each element is arranged as in the claim. As discussed below, none of the rejections based on Stucka's application-built user interfaces come close to meeting these requirements, and thus claims 21-38 are patentable over Stucka as a matter of law.

Regarding the § 102 rejections of claims 21 and 36 based on Stucka, applicants submit that Stucka, which teaches that the application program chooses and builds its user interfaces for display as needed, clearly does not disclose or even suggest an input method, let alone an input method that "includes executable code that draws an input panel in the input panel window" as recited in claim 21, or a selected executable input method that "draws its corresponding input panel by calling a function of the management component" as recited in claim 36. The reasoning proving Stucka's deficiencies is very clear, namely that Stucka's *application* draws the user interface, yet the application of Stucka is unmistakably *not* an executable input method which draws a control panel, since claims 21 and 36 also define another aspect of an input method, essentially that the input method is distinct from the application program. Thus, no reasonable interpretation of Stucka meets the plain claim language with respect to an input method.

Regarding the rejection of claim 23 based on Stucka, applicants submit that Stucka's keyboard (labeled 26 in FIG. 1) cited against applicants is a conventional *hardware* keyboard, not a displayed image of a keyboard as plainly recited. With respect to claim 24, applicants note

In re Application of TOEPKE et al.
Serial No.: 10/072, 111

that column 8, lines 40-53 of Stucka, cited against claim 24, refer to an application's actions in modifying the appearance of a user interface, which is not related to the claimed subject matter.

Regarding the rejection of claim 25 based on Stucka, there is no mention of the claim language of this claim (e.g., regarding focus) in the Office action, nor were applicants able to find anything in Stucka remotely related to such a concept. Applicants respectfully request that each of the claims be examined based on their claimed limitations, and then rejected or allowed, with some reasoning and a proper citation with respect to the claim language provided for any rejection, so that applicants can fairly address the rejection.

The same lack of any specific reference to the claim language also appears (at least) in the rejections of claims 27-33. For example, the Office action has simply rejected claims 22-35 as a whole (based on largely irrelevant drawings and text passages in Stucka), but completely disregarded the language of plainly recited limitations, such as docked versus floating status, a flag, communications between the input method and the application, a size or position of the window, COM objects, and others. Applicants are thus unable to fairly review and comment on these vague and unexplained rejections, except to note that both Stucka and the Office action are silent with respect to these recited limitations.

In sum, Stucka, which teaches that application programs build their own user interfaces, clearly fails to disclose or suggest an input method that draws a control panel and is distinct from the application program. Stucka simply does not disclose these limitations, let alone the elements as arranged as in the claim, and thus fails to support an anticipation rejection of the claims as a matter of law. Further, the rejections are unsupported and/or improper for additional

In re Application of TOEPKE et al.
Serial No.: 10/072, 111

reasons. Reconsideration and withdrawal of the rejections of claims 21 and 23-37 based on Stucka is respectfully requested.

Turning to the §103(a) rejections of claims 1-20 and 39-49, in order to establish *prima facie* obviousness of a claimed invention, by law, all of the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). In addition, “all words in a claim must be considered in judging the patentability of that claim against the prior art.” *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970). Further, if prior art, in any material respect teaches away from the claimed invention, the art cannot be used to support an obviousness rejection. *In re Geisler*, 116 F.3d 1465, 1471, 43 USPQ2d 1362, 1366 (Fed Cir. 1997).

Applicants submit that not only do Stucka and Kono fail to disclose the recited limitation of an interchangeable input method, but moreover submit that if anything, both Stucka and Kono directly teach away from such a concept. For example, Stucka teaches the desirability for an application to construct its own windows and change them as needed, as discussed above, rather than receive its input from anything even resembling an independent, interchangeable input method. The only thing interchangeable in Kono is an insertable RAM IC memory card, thus teaching the need to change hardware to change memory contents.

The deficiencies of Stucka are discussed above, and will not be repeated herein for purposes of brevity, except to reiterate that in Stucka, the application program is intimately in control of its user interfaces, as the application program chooses what it wants to display according to its needs, causes the user interface to be built as desired, interprets events based on how it built the user interface, and so on. No reasonable interpretation of Stucka’s application-

In re Application of TOEPKE et al.
Serial No.: 10/072, 111

built user interface comes close to an input method as defined and claimed in the present invention, let alone the claimed arrangement between the input method and the application programs.

Regarding Kono, the Office action contends that “Kono discloses in figures 7 and 10 that an interchangeable software component and interchangeable hardware component are (21A) configured *[sic]* accept the user data input from an input device associated with the computer system (see figure 7, column 8, lines 28-37 and column 9, lines 37-45).” This contention cannot be reasonably supported, because among other significant differences, there is simply no interchangeable *software* component disclosed in Kono, let alone any teaching or suggestion of an interchangeable *input method* corresponding to that alleged software component. A hardware memory card is not interchangeable software, and a memory card that plugs into a slot on a computing device does not reasonably disclose or suggest an interchangeable software component, let alone a selectable, interchangeable and executable software input method that receives user data input from an input device and communicates with a management component to pass information about the received user data thereto (e.g., claim 1). Indeed, such a memory card is wholly unrelated to an input method as recited in the claims, and it is clearly unreasonable to reject a claim by focusing on one term such as “interchangeable” while completely ignoring the other significant differences of such an element. In fact, a manually pluggable memory card does not come close to reasonably suggesting or providing any motivation for a software input method, let alone one that is “an interchangeable and executable software component that is distinct from the application programs” (see e.g., claim 1).

In re Application of TOEPKE et al.
Serial No.: 10/072, 111

Thus, even if somehow permissible to combine Kono with Stucka (which applicants submit it is not, as discussed below), the claim limitations are not found by such a combination. Further, Stucka and Kono both teach away from the claimed invention, and thus cannot be used to support an obviousness rejection as a matter of law. For at least these reasons, applicants submit that the rejections of claims 1-20 and 30-39 are improper as a matter of law, and respectfully request reconsideration and withdrawal of these rejections.

Also as a matter of law, obviousness may not be established using hindsight obtained in view of the teachings or suggestions of the applicants. *W.L. Gore & Assocs., Inc. v. Garlock, Inc.*, 721 F.2d 1540, 1551, 1553, 220 USPQ 303, 311, 312-13 (Fed. Cir. 1983), *cert. denied*, 469 U.S. 851 (1984). To guard against the use of such impermissible hindsight, obviousness needs to be determined by ascertaining whether the applicable prior art contains any suggestion or motivation for making the modifications in the design of the prior art article in order to produce the claimed design. Even the mere possibility that a prior art teaching could be modified or combined such that its use would lead to the particular limitations recited in a claim does not make the recited limitation obvious, unless the prior art suggests the desirability of such a modification. See *In re Gordon*, 733 F.2d 900, 902, 221 USPQ 1125, 1127 (Fed. Cir. 1984).

As discussed below, the prior art contains nothing at all that supports the conclusion that any of the claims are obvious. In the present case, it is clearly evident that the Examiner can only have used impermissible hindsight gleaned from applicants' own teachings to locate a Stucka and Kono, and thereafter attempted to fit their teachings into applicants' invention, even though neither teach input methods, and both generally teach away from the present invention as claimed.

In re Application of TOEPKE et al.
Serial No.: 10/072, 111

In fact, the alleged motivation in the Office action for combining Stucka with Kono presupposes that Kono teaches an interchangeable software component, is generally nonsensical, and is not found in the prior art. In making the rejection, the Office action alleges that, “it would have been obvious to one of ordinary skill in the art at the time the invention was made to provide an interchangeable software component of Kono to the system display of Stucka et al. as taught by Mori et al. because this would alternatively operate another input device and help the operation of the individual application.” Applicants strongly disagree with this unsupported, broad conclusory statement, as to the extent it is understood, this statement indicates a fundamental misunderstanding of the present invention. Again, not only do Stucka and Kono (as well as Mori et al., presumably U.S. Patent No. 5,644,339) fail to disclose, suggest or provide any motivation for an interchangeable software component (as discussed above), but further, the present invention does not operate other input *devices*, nor do Stucka and/or Kono. It is unclear from this statement how operating other input devices would lead one of ordinary skill in the art to combine these references, or why this might be desirable, or what might be accomplished thereby, let alone how the present invention could somehow result from such a combination.

Instead, it appears as if the Office action located references that are substantially unrelated to the present invention, in an attempt to piece together applicants’ invention using applicants’ own teachings. However, (in addition to failing to reach the claims even when impermissibly combined), such a hindsight reconstruction based on applicant’s teachings is clearly impermissible by law. For at least these additional reasons, applicants submit that the claims of the present invention are patentable over the prior art of record including Stucka and/or

In re Application of TOEPKE et al.
Serial No.: 10/072, 111

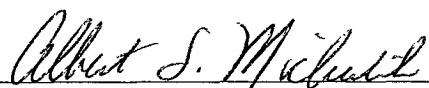
Kono (and/or Mori et al.), and respectfully request withdrawal of the §103(a) rejections of claims 1-20 and 39-49 based thereon.

CONCLUSION

Applicants have reviewed the prior art of record. The claims have been, and continue to be clearly patentable over the prior art of record. In view of the foregoing remarks, it is respectfully submitted that claims 1-21, 23-37 and 39-49 are in good and proper condition for allowance. Entry of the foregoing Amendment and withdrawal of the pending rejections are respectfully requested.

If in the opinion of the Examiner a telephone conference would expedite the prosecution of the subject application, the Examiner is invited to call the undersigned attorney at (425) 836-3030.

Respectfully submitted,



Albert S. Michalik, Registration No. 37,395
Attorney for Applicants
Law Offices of Albert S. Michalik, PLLC
704 - 228th Avenue NE
Suite 193
Sammamish, WA 98074
(425) 836-3030 (telephone)
(425) 836-8957 (facsimile)

In re Application of TOEPKE et al.
Serial No.: 10/072, 111

Appendix A

(marked up copy of the claims amended herein)

18. (Amended) The system of claim 12 wherein the executable input method comprises a [COM] component object model object.

21. (Amended) A system for receiving user data input into a computer system having a plurality of application programs, comprising:

a plurality of executable input methods, each executable input method being distinct from the application programs and other executable input methods and configured to accept the user data input from an input device associated with the computer system, each executable input method further including a defined interface set including at least one interface therein to make the executable input method pluggable into other executable code that is capable of interfacing with the defined interface set;

an input panel window on a touch-sensitive display screen that is distinct from a window of the application, and wherein the executable input method includes executable code that draws an input panel in the input panel window; and

a management component that is capable of interfacing with the defined interface set, the management component being distinct from the application programs and configured to:

- 1) identify one of the executable input methods as a selected executable input method,
- 2) activate the selected executable input method,

In re Application of TOEPKE et al.
Serial No.: 10/072, 111

- 3) communicate with the selected executable input method via the defined interface set,
 - 4) identify information about user data received by the selected executable input method, and
 - 5) pass the information about the received user data to an active application program of the plurality of application programs.
23. (Amended) The system of claim [22] 21 wherein the input panel includes an image representing a keyboard having a plurality of keys thereon.
24. (Amended) The system of claim [22] 21 wherein the management component selectively displays and hides the display of the input panel window.
25. (Amended) The system of claim [22] 21 wherein interaction with the input panel does not cause the input panel window to receive focus.
29. (Amended) The system of claim 21 wherein the executable input method comprises a [COM] component object model object.
36. (Amended) A system for receiving user input data into a computer system having a graphical windowing environment, comprising:

In re Application of TOEPKE et al.
Serial No.: 10/072, 111

an input-output mechanism including a display screen that outputs images and a touch sensitive input device that detects user interaction therewith;

a management component operatively connected to the graphical windowing environment and configured to create an input panel window for display thereof by the graphical windowing environment on the screen, the management component further configured to receive user data and communicate the user data to the graphical windowing environment;

a plurality of executable input methods, each executable input method including a defined interface set comprising at least one interface for calling functions of the management component, and further including an input panel corresponding thereto; and

a selected executable input method of the plurality that:

1) draws its corresponding input panel by calling a function of the management component via the defined interface set,

2) receives user data input via the input panel, and

3) calls a function of the management component via the defined interface set to pass the user data thereto; and

an executable program distinct from the selected executable input method, the graphical windowing environment communicating the user data to the executable program.